



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/727,169	12/02/2003	Michael L. Kazar	SPIN-5	5977
<div>7590 05/02/2007</div> <div>Ansel M. Schwartz Suite 304 201 N. Craig Street Pittsburgh, PA 15213</div>				
			EXAMINER DOAN, DUC T	
			ART UNIT 2188	PAPER NUMBER
			MAIL DATE 05/02/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/727,169

Applicant(s)

KAZAR ET AL.

Examiner

Duc T. Doan

Art Unit

2188

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 March 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-39 and 41-64 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 36 and 37 is/are allowed.
- 6) ☒ Claim(s) 1,3-35,38,39 and 41-64 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Status of Claims

Claims 1-39 have been presented for examination in this application. In response to the last office action, claims 2 and 40 have been canceled, claims 42-64 have been added. As the result, claims 1,3-39,41-64 are pending in this application.

Claims 36-37 are allowed.

Claims 1,3-35,38-39,41-64 are rejected.

Applicant's amendments have overcome the claim objections in previous office action.

Applicant's remarks filed 3/9/07 have been fully considered but they are not persuasive.

Therefore, the rejections from the previous office action are respectfully maintained, with changes as needed to address the amendments.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1,3-35,38-39,41-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fridella et al (US Pub 2005/0044080); incorporating reference (Vahalia's et al (5893140) and further in view of Carns et al (PVFS: A Parallel File System for Linux Clusters).

As in claim 1, Fridella describes an apparatus for data storage comprising: a cluster of NFS (NETWORK FILE SERVER) servers, each server having network ports for incoming file system requests and cluster traffic between servers (Fridella's Fig 1: #110 network file sever, paragraph 27 clustering of data movers #115-#117 provides parallelism as front end to arrays of disks Fig 1: #114 disks);

each server has a network element and a disk element (Fridella's Fig 1, each server #115 must include a network interface element, and a disk/file system interface element; paragraph 28 further discloses each server can be a primary data mover for a disk/file system element; Similarly Friedella teaches by incorporating Vahalia's reference, a network of servers, each server acts a front end (corresponding to the claim's front end network element) for its backend storage disk element such that data in disks can be provided by these servers to clients in parallel manner (Vahalia's Fig 2, Fig 3 show data blocks which are stripped across disks of the disk array are provided to clients in parallel data stream servers in Fig 2 and Fig 3: channel director/disk director elements; Vahalia's Fig 5, column 7 lines 11-20 further discloses software structures such as common file system, physical file systems in each server, allowing servers to communicating with each other, by network means, to keep track data blocks in file systems which are being distributed across servers);

the servers utilizing a striped file system for storing data (Vahalia's Fig 5, column 7 lines 61-66 further discloses data blocks which are stripped across disks in the disk array are being

tracked by physical file systems in each server, and a virtual common file system having maps to map client's file system requests using vnode (virtual node identifier) into the physical node (physical server identifier) in which the data blocks of these file system requests are physically located. Using this virtual node/physical node mapping and logical block address of the client request, the determination of which servers and data blocks serving this client's request can be generated quickly, particularly in the instant of a client's write request, the corresponding data blocks being modified will be immediately flagged (lock flags). This method (i.e mapping vnode/physical node, and maintaining meta data, files attributes of the file systems in primary and secondary movers) will solve the NSF file accessing issue that is normally accesses on a file basis, and thus providing alternative way to access data on block basis (see Vahalia's column 10 lines 1-33 and Friedella's paragraph 40). Therefore, Friedella clearly teaches a method of data blocks in file systems being stored across disks in the disk array, accessing these blocks in an efficiently parallel manner by multiple secondary movers, and by simply tracking "locking" or reserving precisely only data blocks being services in these secondary movers. Friedella and Vahalia do not expressly use the word "stripped". However, in a similar manner, Carns discloses a parallel virtual file system, PVFS, in which data blocks of the file system can be stripped across the I/O nodes/disks (see Carns's Fig 1, section 3.1 second and third paragraphs, trip size is 65 Kbytes). It would have been obvious to one of ordinary skill in the art at the time of invention to include the stripping data blocks of PVFS file system as suggested by Carns in Friedella's system thereby further increasing system data throughput because multiple servers I/O nodes can access data blocks in parallel read/write operations (Carns's section 3.1 paragraph 4).

It's further noted that by stripping data blocks of the file system across several disk elements inherently providing bandwidth to multiple disk element as claimed.

Carn further discloses that the striped file system is comprises a set of local file systems in the stripping i/o nodes, that corresponding to the claim's striped VFSes (see Carn's section 3.2 second paragraph), being distributed among a number of disk elements of the cluster of servers, with one VFS of the set of striped VFSes per disk element (i.e Carn's a local file system of the set of striped local file systems, each local file system per disk element, see Carn's Fig 1 iod0, iod1, etc), wherein a data file is striped among all of the VFSes of the set of striped VFSes with different strips of the file's data in different VFSes in the set of the striped VFSes (see Carn's Fig 1 wherein data file is stripped among all local file systems iod0 to iod5 with different strips of the file data in different VFESes, Fig 1 local/f1092157504 of iod 2 local/f1092157504 of iod 3 er5c..). It is further noted that the local file system in each i/o node typical is a virtual file system so that user would not have to manage the mapping of the data blocks to the physical storage device. Vahalia teaches such a virtual file system VFS to interface and mapping data to physical device including disk tape etc.. (see Vahalia's software Fig 4: #60, column 7 lines 1-28).

As in claims 3, the claim recites wherein each disk element has a virtual file system with the virtual file system of each disk element together forming a striped VFS (Vahalia's Fig 5, column 10 lines 34-36 discloses that client can issues multiples requests to multiple servers for data blocks belonged to files, because data blocks are being stripped across these servers/disks. (distributed loading to servers, granularity down to logical data block level).

As in claim 4, the claim recites wherein all disk elements for a virtual file system act as meta-data servers (Fridella's paragraph 28 discloses that each file system is managed by one data

Art Unit: 2188

mover, primary mover, (corresponding to instant's application meta data server). And the role of being a primary mover can be assigned to any server, for example, Fridella's Fig 1 shows server #111 being a primary mover for file system #121, and server #117 being the primary mover for file system #122).

As in claim 5, Fridella discloses wherein a file has attributes and each server for each file maintains a caching element that stores a last known version of the file attributes and ranges of modification time and change time values for assignment to write operation results (Fridella's paragraph 32 describes file attribute are cached in both primary mover and secondary movers; Fridella's paragraph 38 further describes the secondary movers maintaining/storing the latest time, using a local value m which obtained from the primary clock and value of a local timer/counter for ranges of modification time values of requests being received in these secondary movers).

As in claim 6, the claim recites wherein each disk element, which is not the meta-data server for a virtual file system, is an input output secondary (Fridella's paragraph 28 further discloses other servers are assigned to move only data, functioning as secondary data movers).

As in claim 7, Fridella discloses wherein ranges of file modification times or file change times are reserved from the meta-data server by the input output secondary (Fridella's paragraphs 36-38 describes the update time is a function of the clock time obtained from the clock of the primary mover which including a value m obtained from the primary clock).

As in claim 8, Fridella discloses wherein the modification and change times in the ranges obtained from the meta-data server are issued to operations already queued at the input output secondary. The claim rejected based on the same rationale as of claim 7. Fridella's paragraph 38

Art Unit: 2188

further describes that the secondary movers maintain clock time values using timers/counters and issuing these values for requests in each file's range that it has opened "reserved" for asynchronous write access.

As in claim 9, the claim recites wherein modification and change times in the ranges obtained from the meta-data server are issued to operations received during a window of time after the ranges are reserved from the meta-data server by the input output secondary (Fridella's paragraph 46 discloses that after a first secondary mover obtaining a reserved time from the primary mover for the first asynchronous write for a range data blocks, while executing subsequent write requests for these data blocks (and before committing these data blocks), for example, a second secondary mover may obtain another reserved time for a second set of data blocks; due to clock skewing among different servers (Fridella's paragraph 33, clocks of servers are not synchronized), the first secondary mover must compare these time values and adjust its local cached time value accordingly (also see Fridella's paragraph 12, clock time value from the first mover, time interval measured by the secondary mover)).

As in claim 10, the claim recites wherein operations affecting all stripes of a file begin executions first at the meta-data server for a file and then execute at all input output secondaries, such that operations at the input output secondaries wait only for already executing operations that have already finished their communication with the meta-data server. (Fridella's paragraph 11, all secondary movers's write requests must begin by communicating with the primary mover to obtain meta data values including a modified time value at the first asynchronous write request. Subsequent second asynchronous write requests which are directed to each secondary mover, are handled by the secondary movers without the need to communication with the

primary movers. As such, these second asynchronous write requests can be understood as “already finished their communication with the primary mover”, as claimed.

As in claim 11 the claim recites an apparatus as described in claim 10 wherein operations follow one of at least two locking models, the first of which is to synchronize first with the meta-data server, then begin core execution by synchronizing with other operations executing at the input output secondary, and the second of which is to first synchronize at the meta-data server, and then to synchronize with operations at one or more input output secondaries that have begun core execution at the input output secondaries. The claim rejected based on the same rationale as in claim 10. The locking models merely indicate means to reserve a range of data blocks for modifications such that integrity of these data blocks is preserved. Friedella’s paragraph 11 teaches the first method to reserve data blocks by communicating with the first/primary data mover, and the second method done by the secondary data mover for subsequent second asynchronous data requests, synchronizing/ordering these requests among themselves (client’s requests pending in the secondary data mover, waiting for being executed by the secondary mover in an asynchronously manner).

As in claims 12-13, the claims recite wherein the cluster network is connected in a star topology (claim 12; Friedella’s Fig 1, any client can access data using any mover in a star topology); wherein the cluster network is a switched Ethernet (claim 13; Friedella’s paragraph 27 lines 10-16).

As in claim 14, the claim recites creating a file across a plurality of NFS (NETWORK FILE SERVER) servers; writing data into the file as strips of the data in the servers, the strips together forming a stripe; reading strips of the data from the servers; and deleting the strips from

the servers. The claim rejected based on the same rationale as in the rejection of claims 1 and 3. Fridella's column 10 lines 25-30 further discloses meta data structures in servers capable of keeping track of data blocks that are destaged/upstaged from the disk array. Thus they are being used to maintain data of the trips in servers, or deleting the trips from the servers when data are destaged to disks.

Claim 15 rejected based on the same rationale as in the rejection of claim 6.

Claim 16 rejected based on the same rationale as in the rejection of claim 5.

Claim 17 rejected based on the same rationale as in the rejection of claim 7.

As in claims 18-19, including the step of making a status request by the caching element to the meta-data server to obtain a file's current attributes (claim 18;) wherein the making a status request step includes the step of obtaining modification time and change time ranges from the meta-data server (claim 19). Friedella's paragraph 43, secondary mover sends a request to the primary mover to obtain the current meta data attribute values, FmpGetAtr, and receiving modification time value from the primary mover.

As in claims 20,21,23 the claims recite including the step of queuing file read and file write requests at the input output secondary until the file read and file write requests are admitted by the cache element and complete execution (claim 20; Fridella's paragraph 39 discloses each secondary mover can process client's subsequent second asynchronous write requests. In order to processing the requests in asynchronous manner, a queue must be employed to keep these clients's requests and subsequently executing them latter, in an asynchronous manner); including the step of tracking by the cache element of the file read and file write requests executing for the file and the ranges that are being read or written (claim 21; Fridella's paragraph 39, each

secondary data mover tracks the subsequent second asynchronous write requests for the data blocks/range that it has been reserved at the first asynchronous write request); including the step of checking a byte range affected by a file read request to ensure it does not overlap a byte range of any file write requests previously admitted and currently executing (claim 23; Fridella's paragraph 39 clearly suggests the subsequent second asynchronous write requests must be checked using the modification time value to insure these requests on the data blocks are done in proper order as in client's requests);

As in claims 22,24 the claims recite including the step of requesting the cache element move out of invalid node to read mode when a read operation must be executed (claim 22); including the step of requesting, in response to a file write request that the cache element move into a write mode (claim 24). Fridella's Fig 3 #151 to #159 shows a secondary mover receives a file access request that moves it into a corresponding state (read or write accessing mode) and executing sequence of steps to fulfill the request.

As in claim 25, the claim recites including the step of checking with the cache element the byte range affected by the file write request for overlap with any admitted and still executing file read or file write requests. The claim rejected based on the same rationale as in claim 24, that is to protect the integrity of data being updated, each secondary mover must check subsequent second asynchronous requests using the modify time values, so that these requests from clients are processed in proper order as intended by the client.

As in claim 26, Fridella discloses when executing a write request, of allocating a modification time and change time pair from the range of modification times and change times stored in the cache element (Fridella's paragraph 39 discloses when receiving subsequent second

asynchronous write requests, the secondary mover keep tracks of modification times of these request, storing these values in its cache).

As in claim 27, the claim recites including the step of checking the head of a queue of pending file read and file write requests to see if a head request can be admitted by the caching element after either a file read or file write request is completed. The claim rejected based on the same rationale as in the rejection of claim 25.

As in claim 28, Fridella discloses including the steps of detecting by the cache element that a file length must be updated in response to a file write request, moving the cache element into exclusive mode; and making a file write status call to the meta-data server to update length attributes of the file (Fridella's paragraph 39 discloses when the secondary server performing the write commit operation, all previous operations are completed, such that the changed data in this cache can be flushed to disk exclusively, the secondary mover also sends the updated attributes associated with these data blocks/length to the primary server (see Friedella's paragraph 45)).

Claim 29 rejected based on the same rationale as in the rejection of claim 5.

Claim 30 rejected based on the same rationale as in the rejection of claim 7.

Claim 31 rejected based on the same rationale as in the rejection of claim 18.

Claim 32 rejected based on the same rationale as in the rejection of claim 19.

Claim 33 rejected based on the same rationale as in the rejection of claim 22.

Claim 34 rejected based on the same rationale as in the rejection of claim 24.

Claim 35 rejected based on the same rationale as in the rejection of claim 28.

As in claim 38, the rationale in the rejection of claim 1 is incorporated herein. The claim recites a method for reading data in a file comprising the steps of: receiving an NFS (NETWORK FILE SERVER) read request for data in the file at a network element (Fridella's paragraph 27 lines network devices communicate with each other using NFS and CIFS command); determining by the network element which VFS stores at least one strip containing the data (see rationale of claim 1); sending a file read request from the network element to at least one disk element of a plurality of servers storing a strip of the data (Friedella's paragraph 29, client sending access requests to multiple secondary movers); obtaining current attributes associated with the file by each disk element; reading the strips of the file from each disk element having the strips; and generating a response in regard to the file read request (Obviously, the secondary movers must generated a response and returning the read data to client).

As in claim 39 the claim recites a method for writing data in a file comprising the steps of: receiving an NFS (NETWORK FILE SERVER) write request for a file at a network element; determining by the network element which VFS is associated with the file; sending a file write request from the network element to at least one disk element of a plurality of servers having a stripe of the VFS; acquiring current attributes associated with the file; and writing a predetermined number of bytes of the data into each VFS strip in succession until all of the data is written into the file. The claim rejected based on the same rationale as in the rejection of claims 19,38.

As in claim 41, the claim recites identifying a disk element for a virtual file system of an NFS (NETWORK FILE SERVER) server as a meta-data server as a meta-data server and a disk elements for the NFS servers which are not identified as the meta-data server as input output

secondaries. The claim rejected based on the same rationale as in claim 40. Fridella's Fig 1, paragraph 28 discloses a mover #155 is designated as primary mover for a file system A (corresponding to the claim's meta-data server), while other movers are designated as secondary movers with respect to file system A (corresponding to the claim's input output secondaries).

As in claim 42, the rejection of claim 1 is incorporated herein. That is the stripped file systems comprises a set of striped VFSes. The different VFSes (i.e Carn's local file systems) that referencing and identifying to the same striped file system, for example Carn's Fig 1 discloses local file systems referencing identifying the same striped file system f1092215504. Vahalia further disclose that the virtual file system is identified by using its (Vnode number /VFS) interface (see Vahalia's column 7 lines 63-67).

As in claim 43, Carn's Fig 1 discloses the data-strips of the stripped file system are distributed across the set of VFSes as recited in the claim.

As in claims 44-45, Fridella further discloses wherein one server of the cluster of servers is a meta-data server (Fridella' Fig 3: #134) for one of the striped VFSes (claim 44, Fridella's Fig 3: #133); wherein all files of the one of the striped VFSes are represented at the meta-data server (claim 45, Fridella's Fig 3: #134, paragraph 38, primary maintains attributes representing the files at secondary servers).

As in claims 46-47, the claims are rejected based on the same rationale as of claim 43.

Claim 48 is rejected based on the same rationale as of claims 1 and 5.

As in claim 49, Fridella further discloses wherein each disk element which is not the meta-data server is an input output secondary (Fridella's Fig 3: #133 secondary server); Claim 49 is rejected based on the same rationale as of claim 6.

Art Unit: 2188

Claim 50 is rejected based on the same rationale as of claim 7.

Claim 51 is rejected based on the same rationale as of claim 8.

Claim 52 is rejected based on the same rationale as of claim 9.

Claim 53 is rejected based on the same rationale as of claim 10.

Claim 54 is rejected based on the same rationale as of claim 11.

Claim 55 is rejected based on the same rationale as of claim 12.

Claim 56 is rejected based on the same rationale as of claim 13.

As in claim 57, Fridella discloses the servers are NFS servers (Fridella's paragraph 37 lines 1-2).

Claim 58 is rejected based on the same rationale as of claim 48.

Claim 59 is rejected based on the same rationale as of claim 7.

Claim 60 is rejected based on the same rationale as of claim 8.

Claim 61 is rejected based on the same rationale as of claim 9.

Claim 62 is rejected based on the same rationale as of claim 10.

Claim 63 is rejected based on the same rationale as of claim 11.

Claim 64 is rejected based on the same rationale as of claim 57.

Allowable Subject Matter

Claims 36,37 are allowed.

Response to Arguments

Applicant's arguments in response to the last office action has been fully considered but they are not persuasive. Examiner respectfully traverses Applicant's arguments for the following reasons:

Regarding the remark on page 21 second paragraph, it is noted that the new claims have been added in the amendments are claims 42-64.

As to the remarks on pages 21-31 for the rejections of claims 1,3-35, Examiner respectfully disagrees with Applicant's remarks as follows,

A) Fridella discloses a method in which a ranges of data can be requested and reserved by the secondary sever for a duration of time so that the secondary server can access these data blocks. The secondary sever further consult the meta-data server (i.e primary server) which determines the availability of these data blocks and grants the secondary server for accessing these data blocks (Fridella's paragraph 41). It is noted that specification's page 20 third paragraph discloses the same procedure. Therefore, Examiner disagrees with the Applicant's remarks on page 23 first paragraph that somehow the request and granting permission for blocks of data that is so disruptive that Fridella's teaching cannot be used to reserving data blocks for a file system, simply because this procedure is also disclosed in the specification's page 20 third paragraph.

Examiner further notes that the procedure is disclosed in the specification's page 25 second paragraph, that is the secondary I/O doing the first spin write request to obtain a ranges of modification time values from the meta data server. It then uses these time values for accessing

Art Unit: 2188

these data blocks in subsequent 49 spin write requests. Thus only one (the first spin write request) operation is required to contact the meta data server.

Thus, in contrast to Applicant's allegation that Fridella's "the request and granting permission for blocks of data that is so disruptive that Fridella's teaching cannot be used to reserving data blocks for a file system", Fridella clearly teaches the reservation step only required for the first write request, exactly in the same way disclosed in the instant's application.

Fridella's paragraph 38 further discloses that the step obtaining a lock or reserving a range of data blocks is only required for the **first** asynchronous write request of these data block, so that the primary data mover can assign the file modification time values for these data blocks to the secondary mover. For all subsequent second asynchronous write requests, the secondary mover does not have to further contact the primary mover, it can executes these second asynchronous write requests independently and in parallel manner, thus the data throughput of the system is greatly improved, since multiple secondary movers can service multiple second asynchronous write requests for data blocks being stripped across disks in a parallel manner.

It is further noted that the Applicant's arguments on pages 21-25 for the procedure of the secondary sever request and the metadata server granting do not have any bearing on the claim 1's limitations which describe the data being distributed over several servers/storage elements.

B) Regarding the Applicant's remarks on page 30 for the amended recitation "for providing bandwidth to multiple disk elements". It's further noted that by stripping data blocks of the file system across several disk elements inherently providing bandwidth to multiple disk element as claimed.

Carn further discloses that the striped file system is comprises a set of local file systems in the stripping i/o nodes, that corresponding to the claim's striped VFSes (see Carn's section 3.2 second paragraph), being distributed among a number of disk elements of the cluster of servers, with one VFS of the set of striped VFSes per disk element (i.e Carn's a local file system of the set of striped local file systems, each local file system per disk element, see Carn's Fig 1 iod0, iod1, etc), wherein a data file is striped among all of the VFSes of the set of striped VFSes with different strips of the file's data in different VFSes in the set of the striped VFSes (see Carn's Fig 1 wherein data file is stripped among all local file systems iod0 to iod5 with different strips of the file data in different VFESes, Fig 1 local/f1092157504 of iod 2 local/f1092157504 of iod 3 er5c..). It is further noted that the local file system in each i/o node typical is a virtual file system so that user would not have to manage the mapping of the data blocks to the physical storage device. Vahalia teaches such a virtual file system VFS to interface and mapping data to physical device including disk tape etc.. (see Vahalia's software Fig 4: #60, column 7 lines 1-28).

Therefore Carn clearly teaches **the data blocks of the striped file system being stripped across several local file systems in i/o nodes** (see Carn's Fig1 and section 3.2 third paragraph). These data blocks of the stripped file system are easily being mapped in local file systems using the vnode and virtual file interface vnode/VFS as taught by Vahalia (see Vahalia's column 7 lines 62-67). Furthermore these stripped data blocks are being accessed by the file system in parallel and in an efficiently manner by using the meta data server which maintains and grants the permission for accessing these data block to the secondary servers for a duration of time (see Fridella's paragraph 41).

C) Regarding the Applicant's remarks on pages 26-28, Examiner relies on Vahalia for the teaching of how readily the data blocks can be mapped using vnode identifiers associating with several virtual file systems interfaces NFS, VFS etc.. such that the data can be stored and retrieved in the physical storage medium transparently to user's requests. The Vahalia's teaching of moving data blocks from disk array devices/medium to the tape devices/medium consistently and further teach how easily and transparently the striped data blocks for example in the striped disk array devices can be accessed and moving to different devices such as tape devices and vice versa.

D) Regarding the Applicant's remarks on pages 28-29 for the teaching of Carn's section 3.1.

Firstly, Carn clearly defines that in order to provide bandwidth to several disks elements of i/o nodes, the data blocks of the striped file system is being striped across these disk elements (see Carn's section 3.1 first paragraph, lines 9-12 which teaches data blocks of the striped file system are stripping across several physical devices).

Secondly, Carn teaches that typically, the local file systems and the NFS interfaces requires the mounting actions by the system administrators. However, Carn further discloses an enhancement in which these mounting and mapping of local file systems and the NFS interfaces are done in a transparently and automatically manner using some software code in the PVFS library. There is nothing in Carn's teaching that makes the allegation "...Thus Carns specifically teaches against using stripping with NFS.." as Applicant's remarks on page 29.

E) Regarding Applicant's remarks on pages 29-31, specifically on page 29 second paragraph, Applicant's remarks seem to be that just because Carn teaches a striped virtual file

Art Unit: 2188

system wherein data blocks are stripped and accessed in a parallel manner, and somehow it cannot combined with the Vahalia and Fridella teaching since somehow they teaches away from applicants' claim invention. Examiner respectfully disagrees, since Fridella, Vahalia and Carn teaching in combination disclose all limitations as claimed (see discussion in above items A to item D).

F) Regarding Applicant's remarks on page 31-32 for the rejection of claims 38-39 are rejected based on the same rationale as discussed in items A-E.

G) Regarding Applicant's remarks on page 32-35 for the rejection of claims 48, 58,59-64.

As to the claim's limitation "where one disk element for a given file system acts as a meta-data server". The limitation is clearly taught by Fridella's Fig 2: #134 primary NFS server). Carn's section 3.1 fourth paragraphs similarly discloses meta-data being kept in the PVFS manager (corresponding to the claim's meta-data server) so that the PVFS manager can provides meta-data information to the applications.

Regarding Applicant's remarks on page 33 third paragraph for the teaching of Carn's section 3.1, please see item D.

Regarding Applicant's remarks on page 33-35 for caching and storing file attributes such as modify time, Fridella further discloses wherein a file has attributes and each server for each file maintains a caching element that stores a last known version of the file attributes and ranges of modification time and change time values for assignment to write operation results (Fridella's paragraph 32 describes file attribute are cached in both primary mover and secondary movers; Fridella's paragraph 38 further describes the secondary movers maintaining/storing the latest time, using a local value m which obtained from the primary clock and value of a local

Art Unit: 2188

timer/counter for ranges of modification time values of requests being received in these secondary movers). Fridella's paragraph 41 further teaches once the associated attributes obtained (i.e. granted) by the meta-data server, the secondary uses these attributes to access the file for the clients, that is secondary can read, write **directly** to the file system (see Fridella's paragraph 41 lines 15-19). In a consistent manner, Carn's sections 3.1 forth paragraph teaches once the attributes values are provide by the PVFS manager, the application can access data blocks being stripped and stored across several i/o node/servers **directly**.

Claims 58-64 are rejected based on the same rationale as of claim 48 and discussion in above paragraphs.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 36 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

Art Unit: 2188

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

When responding to the office action, Applicant is advised to provide the examiner with the line numbers and page numbers in the application and/or references cited to assist examiner to locate the appropriate paragraphs.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Duc T. Doan whose telephone number is 571-272-4171. The examiner can normally be reached on M-F 8:00 AM 05:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung S. Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.


HYUNG SOUGH
SUPERVISORY PATENT EXAMINER

4-29-07